Unwasted DASE : Lean Architecture Evaluation

Tuovinen, Antti-Pekka

Springer International Publishing AG
2017-11

Tuovinen , A-P , Mäkinen , S , Leppänen , M , Sievi-Korte , O , Lahtinen , S & Männistö , T 2017 , Unwasted DASE : Lean Architecture Evaluation . in M Felderer , D Mendez Fernandez , B Turhan , M Kalinowski , F Sarro & D Winkler (eds) , Product-Focused Software Process Improvement : 18th International Conference, PROFES 2017, Innsbruck, þÿ Austria, November 29 December 1, 2017, Proceedings . Lecture Note Science , vol. 10611 , Springer International Publishing AG , Cham , pp. 128-136 , International Conference on on Product-Focused Software Process Improvement , Innsbruck , Austria , 29/11/2017 . https://doi.org/10.1007/978-3-319-69926-4_10

http://hdl.handle.net/10138/232349
https://doi.org/10.1007/978-3-319-69926-4_10

unspecified
acceptedVersion

Unwasted DASE: Lean Architecture Evaluation

Antti-Pekka Tuovinen[1], Simo Mäkinen[1] Marko Leppänen[2,3], Outi Sievi-Korte[2], Samuel Lahtinen[2], and Tomi Männistö[1]

[1] University of Helsinki, Department of Computer Science, Finland
{antti-pekka.tuovinen@helsinki.fi}
[2] Tampere University of Technology, Department of Pervasive Computing, Finland
[3] Ministry of Education and Culture, Finland

Abstract. A software architecture evaluation is a way to assess the quality of the technical design of a product. It is also a prime opportunity to discuss the business goals of the product and how the design bears on them. But architecture evaluation methods are seen as hard to learn and costly to use. We present DASE, a compact approach that combines carefully selected key parts of two existing architecture evaluation methods while making evaluation lean and fast. We have applied DASE in three industrial cases and the early results show that even a one-day evaluation workshop yields valuable results at a modest cost.

Keywords: Software architecture evaluation, Product development

1 Introduction

The purpose of *architecture evaluation* is to analyze how well a software system will satisfy its quality requirements, uncover the key architectural decisions, and identify risks. The effectiveness of architecture evaluation has been proven many times after the introduction of the first methods in the late 1990's [1–4].

An evaluation is an *expert review* that requires the key technical persons and other stakeholders to meet in person and contribute their knowledge. Usually, this entails a deep discussion about the mission of the product and about the possibilities created by technology. This strengthens common understanding of the goals but provides also a golden chance to share experiences, knowledge, and the rationale behind the design decisions [3] [2, p. 6]. These 'soft effects' may in practice be even more valuable than the hard technical results [5].

However, architecture evaluation is still not yet a common practice in the industry [6, 7, 5]. There is a learning curve, a general perception of high cost, and problems in quantifying results for decision making, which make companies reluctant to adopt them [1, 3, 8]. Industrial use is reported, e.g., in [4, 5, 1, 7, 3].

An architecture evaluation is typically a staged review process. Depending on the method, the different stages produce outputs that are not necessarily used later on. For example in some scenario-based methods, due to time pressure and set priorities, half of the created and elaborated scenarios may not be used at all [1], which is considerable waste. Also, a perceived dependency on explicit architecture design may alienate agile developers. They fear that "big design up-front" incurs waste because if the design assumptions are incorrect or out-of-date, it leads to inability to cope with change and to expensive rework [9].

In this paper we present our architecture evaluation approach called DASE, short for decision and scenario based architecture evaluation. Our goal is to make architecture evaluation more attractive to practitioners: First, based on our own experiences, we have carefully selected from well-known scenario-based and decision-centric methods the elements that we know to work best. Second, we aim at minimizing the calendar time and the resources needed for an architecture evaluation by concentrating the main effort in a one-day workshop and by keeping the learning curve low. Third, we keep the evaluation focused on the most important things from both business and technology viewpoints.

In Section 2, we discuss existing architecture evaluation methods and our own experiences. Section 3 describes the DASE approach. We describe three industrial case studies of using DASE and the results, observations, and feedback in Section 4. Section 5 presents the conclusions and outlines future work.

2 Background

2.1 Architecture Evaluation Methods

The idea of scenario-based methods, like ATAM [10, 11], is to evaluate an architecture through *scenarios* elicited from all *stakeholders* in workshops. Typically, a scenario focuses on one quality aspect from a quality tree (defined in the process) and specifies a situation and a stimulus that test the response of the system and its architecture. The scenarios are evaluated to determine the system's response and to identify risks. To be useful, scenarios need to be concrete, clear and prioritized. The results of properly executed assessments are valuable and usually well received [1, 3]. Scenarios are a powerful tool for assessing the adequacy of the system under evaluation and also for making the technical people aware of the needs of the business and for making the business people aware of the technical opportunities and challenges [1, 3].

Instead of scenarios, the Decision-Centric Architecture Review method DCAR [5] focuses on identifying *architectural design decisions*, their rationale, and their interrelationships. The decisions are ranked based on importance. In the evaluation part, the participants discuss the forces affecting the decisions and their consequences (pros and cons) and vote whether each decision is good or needs to be reconsidered. The importance of identifying and analyzing the key design decisions is also emphasized by others [7]. Because of their emphasis on design and its consequences, these kind of methods could be characterized as bottom-up or inside-out as opposed to scenario-based methods that emphasize requirements.

Other approaches include, e.g., the TARA method [7] that is a light-weight expert review where a single assessor does the evaluation (consulting others) according to a specific focus of interest. At the other end of the spectrum is the comprehensive RATE approach [1, 2] that recognizes different types of assessments based on their purpose and employs several analysis and evaluation methods including ATAM, as needed.

On the down side, an architecture evaluation typically requires several meetings, couple of weeks of calendar time, and tens of person hours (sometimes hundreds) [1]. Understandably, organizations may be reluctant to make this investment [7]. Also, there is a certain learning curve [3, 7]. For example, the construction of the quality tree and the formulation and prioritization of the quality attribute scenarios in ATAM can be challenging[3, 12, 1]. The results can sometimes be hard to quantify for decision makers [1].

2.2 Our Architecture Evaluation Experience

Tampere University of Technology has a lot of experience in architecture evaluation [3, 8]. At TUT, the third author facilitated about ten evaluations carried out in the local industry, using either ATAM [11] or DCAR [5]. The first author has over ten years of experience in architecture work in mobile device industry including architecture evaluations. Several of the authors have also experience in teaching the methods. The evaluations have brought a lot of insights. First, companies are rarely willing to invest in an evaluation – except in the engineering domain, where architecture is valued and changes are slow [3]. Second, the most valuable outcome was information transfer: both the stakeholders and the architect gained valuable knowledge. For example, scenario creation was often turned into an ad hoc requirements elicitation workshop. Last, but not least, the architects felt that they had been designing the system on their own and the evaluation provided an opportunity to challenge their decisions. Usually, the architect was the 'defendant' and had to explain the rationale of a decision. Even if the other participants did not have the expertise to really challenge it, the questions posed and the process of explaining the decision deepened understanding and forced the architect to see the problem from different perspectives. So, the assessment acted as a form of 'rubber duck debugging' [13, p. 95].

3 Decision and Scenario-Based Architecture Evaluation

The DASE approach picks and combines parts of architectural design decision review from DCAR [5] and parts of scenario analysis from ATAM [11]. The idea is to work faster by involving less people and by focusing on key issues. There are two phases: First, in the pre-work (preparation) phase, technical and business information is collected and processed into a list of decisions and scenarios. Second, the decisions and scenarios are used to evaluate the architecture in a one day workshop (the main phase). Figure 1 shows an overview.

3.1 Pre-work

The pre-work phase is centered around an interview (1–2 hours) where the *product owner* presents the mission and the business objectives of the product and

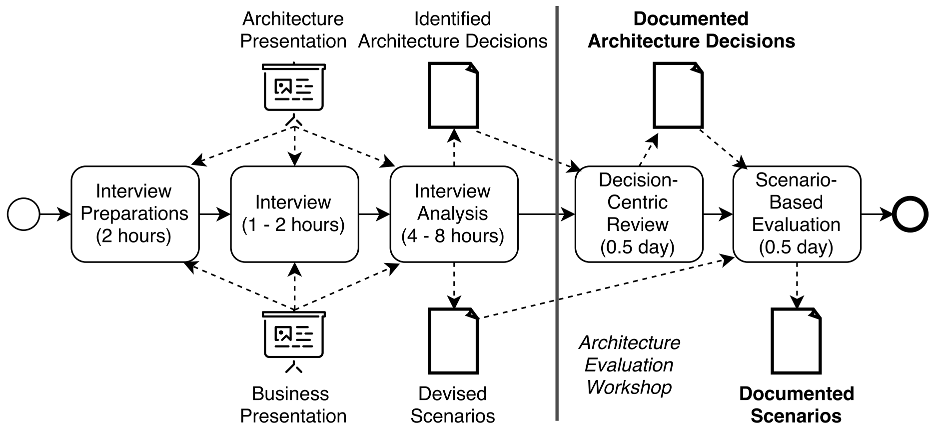

Fig. 1. Phases in the decision and scenario-based architecture evaluation (DASE)

the *architect* presents the technical solution. The interview is lead by the *facilitators*, a team of two to three people who facilitate and guide the evaluation process from start to finish.

Based on the presentations, their notes (and possible recordings), the facilitators identify and list significant design decisions (described in a few words). In this point the design starts to become visible, if it wasn't so before. The architect is asked to check the list. Together with the architectural information, the business objectives are then used to devise a number of scenarios that reflect the interests and topical issues that came up in the interview. Good scenarios are relevant to the stated business objectives and challenge the architecture in some way, by targeting potentially problematic areas or by focusing on architectural hot spots that could affect system responses. The facilitators have an important role in identifying the key issues because there is no extensive scenario collection and elaboration phase involving multiple stakeholders. The scenarios are reviewed in the next phase and can be adjusted, if needed.

3.2 Architecture Evaluation Workshop

The one-day architecture evaluation workshop is split into two sessions: the first (morning) is for the decision-centric review and the second (afternoon) for the scenario based evaluation. The order is important because the design is then fresh in mind when evaluating the scenarios. In addition to the *facilitators*, the morning session requires the presence of the *architect* and other technical people (who know the design) and *business stakeholders* are required for the afternoon session. The presence of the *product owner* is good also for the morning session.

Decision-Centric Review The objective of the morning session is to evaluate and document the most important architectural decisions. The session follows the general protocol of DCAR, leaving out the analysis of the design forces [5].

In the beginning, the *facilitators* present the list of decisions they have identified earlier. Next, the participants vote for the most important decisions. Each participant has a pool of votes (e.g. 10 voting points) to assign freely to decisions; the facilitators can also vote but with fewer points (e.g. 5). The decisions are then ordered by the vote count so that those with the highest count are selected for review. The selected decisions are documented with a fixed template [5] that is used to explain the issue solved by the architecture decision, potential alternative solutions, the rationale for the particular solution, and possible drawbacks. The facilitators can document the decisions together with the developers or the developers can do it themselves, working in parallel for more efficient use of time. A pre-filled example helps developers to get started. Once the selected decisions have been documented, each decision is discussed together. The discussion ends with another vote where the participants are asked whether the decision can be considered a good one, neutral, or risky. The number of decisions documented depends on the number of people working on them, typically it is three to five.

Scenario-Based Evaluation The afternoon session focuses on the scenarios. The scenarios have been defined using the ATAM [11] template that includes a descriptive name, the related quality attributes, an environment description, the stimulus triggering the scenario, and the expected response.

First, the facilitators present an overview of the devised scenarios in order to assess their feasibility and to make any adjustments. Each scenario is then evaluated so that the participants will try to explain what would happen when the triggering event happens, given the conditions, and whether the expected outcome (system response) would be achieved. The facilitators add the explanation to the scenario's description. The architectural decisions affecting the outcome are also noted and listed. The decisions have just been reviewed, which helps in this. The workshop ends when all relevant scenarios are covered or when time runs out. At this point, the facilitators wrap up the workshop and ask for immediate feedback. The facilitators supply a report of the *documented decisions* and *scenarios* to the participants after some final editing. These reports are the concrete outcome of the evaluation.

4 Applying the Method

The DASE method has been tested in three Finnish companies during 2015–2017. The companies were selected by convenience and based on their own interest. Two of the companies, A and B, were participants of the Finnish software research program Need for Speed[1]. Each architecture evaluation focused on a single project. The first, second and third author were the facilitators in cases A and C, and the first, second, and fourth author in B.

As general results, in all cases the company representatives found the evaluations useful and they appreciated the concise schedule. Below, we describe the cases and the findings that are specific for them. The number of decisions and

[1] http://www.n4s.fi/en

scenarios in each case are listed in Table 1. The amount of covered scenarios in all cases was about 10 (same as reported, e.g., in [14]).

Case A was a mid-sized software company working in business software solutions. The evaluated project was a large and mature accounting product. Architecturally, they needed to take into account multiple customers sharing same server-side data resources and many external dependencies (banks, tax office).

The four documented decisions concerned the customer database design (single-tenancy), the nature of transactions in the services, the requirements for strong authentication in many system functions, and the technical infrastructure of the application hosting servers and database servers. The scenarios touched on quality attributes such as maintainability, availability, and scalability of the system that were affected by the design decisions on data management.

The participants of the evaluation workshop found the scenarios mostly relevant but thought that some of the scenarios were too exploratory and unlikely in the near future. A new employee was present in the workshop and he thought it was good training for him.

Case B was a small company developing an on-demand video broadcasting application for different devices. The product was not very big but there were several versions of it, and it was already in the market. The architecture of the product had been just restructured to be more modular and flexible to enable parallel development of features. There were also real-time requirements.

The three documented decisions defined the key aspects of the new plug-in architecture that promoted separation of concerns and made testing easier, for instance. Modifiability and testing were the main themes in scenarios. Scalability did not concern company B because the customer is responsible for the infrastructure. Two new scenarios were created on the fly in the afternoon session.

The mixing of 'bottom-up' (decisions) and 'top-down' (scenarios) analysis was appreciated, and the order of the sessions was considered good. However, even better scenarios could have been devised if the facilitators could have used the product first. It was seen helpful to have architecture decision templates with concrete examples in order to get started with documenting the decisions.

Case C was a mid-sized company closely associated with the public sector in Finland. The product evaluated was a recently launched user authentication solution meant for educational on-line services for primary schools. The product was developed by a single person. Company C wanted to utilize and maintain an open-source based solution for its product in an environment where there are many authentication providers and learning service providers.

The three documented decisions concerned the use of an open source authentication framework as a basis for the solution, storing of client configurations in databases, and a specific dependency to legacy code that added extra complexity to the overall architecture. The scenarios touched on the central role of the company acting as a hub and a connection point for authentication and service providers. Interoperability and maintainability were important quality attributes as well as the ability to integrate new providers. Being able to monitor the responsiveness of the authentication providers was also important.

Table 1. Number of decisions and scenarios in evaluation cases

	A	**B**	**C**
Decisions identified	20	16	9
Decisions receiving votes	15	11	6
Decisions documented	4	3	3
Scenarios devised	10	15	10
Scenarios evaluated	10	13	8
Scenario waste ratio	0%	13%	20%

The architect would have liked the facilitators to more strongly challenge the solution and provide alternatives. Doing evaluations regularly as part of development was considered possible but it was seen important to get an external viewpoint. Reporting and making the outcome of the scenario analysis actionable was also raised as a topic as people were uncertain what to do with the scenarios after the evaluation. One suggestion was that the evaluation report could include options and recommendations for addressing a particular concern.

5 Conclusions

We have presented the DASE approach that combines selected activities from two best of breed architecture evaluation methods into a compact process. DASE has been validated in three commercial projects, and the results show that at a modest use of resources (two days per facilitator and one day per other participant) an architecture can be successfully evaluated in a one-day workshop. The participants saw the evaluations as useful in general and appreciated the broad perspective to architecture they gained through design decisions and scenarios. The facilitators do need to understand design decision analysis and scenario evaluation enough to guide the process, but templates and pre-filled examples help participants to get quickly on board. The facilitators have a key role in preparing for the main evaluation workshop and in keeping it focused on key issues. However, based on our earlier experiences, the facilitators have even more coaching and guiding to do when doing an ATAM-evaluation, for example.

As criticism and improvements, some participants asked for more actionable results that would guide further development. They asked for challenging the design stronger and for suggesting alternative solutions. This implies that the goals of an evaluation need to be openly discussed and that relevant expertise must be available, e.g. an internal consultant from another team. We observed also the risk that because the participants select the decisions to document, there may be a tendency to select only ‘good’ decisions. In the three cases, none of the documented decisions were considered problematic. The situation might be different for a system in an early phase of development.

As further work, it would be important to study how the approach scales up for really big systems. Also, the consequences of the fact that the number of documented design decisions seems to be constant need to be understood better.

Acknowledgments

This work was supported by the Finnish Funding Agency for Innovation (Tekes) as part of the N4S Program of DIMECC[1].

References

1. Knodel, J., Naab, M.: Software architecture evaluation in practice: Retrospective on more than 50 architecture evaluations in industry. In: Software Architecture (WICSA), 2014 IEEE/IFIP Conf. on. (April 2014) 115–124
2. Knodel, J., Naab, M.: Pragmatic Evaluation of Software Architectures. The Fraunhofer IESE Series on Software and Systems Engineering. Springer (2016)
3. Reijonen, V., Koskinen, J., Haikala, I.: Experiences from scenario-based architecture evaluations with ATAM. In Ali Babar, M., Gorton, I., eds.: Software Architecture, 4th European Conference on (ECSA 2010). Volume 6285 of Lecture Notes in Computer Science. Springer (2010) 214–229
4. Bellomo, S., Gorton, I., Kazman, R.: Toward agile architecture: Insights from 15 years of ATAM data. IEEE Software (5) (2015) 38–45
5. van Heesch, U., Eloranta, V.P., Avgeriou, P., Koskimies, K., Harrison, N.: Decision-centric architecture reviews. Software, IEEE **31**(1) (January 2014) 69–76
6. Bass, L., Nord, R.: Understanding the context of architecture evaluation methods. In: 2012 Joint Working IEEE/IFIP Conf. on Software Architecture (WICSA) and European Conf. on Software Architecture (ECSA). (August 2012) 277–281
7. Woods, E.: Industrial architectural assessment using TARA. Journal of Systems and Software **85**(9) (2012) 2034–2047
8. Eloranta, V.P., Koskimies, K.: Lightweight architecture knowledge management for agile software development. In Ali Babar, M., Brown, A.W., Mistrik, I., eds.: Agile Software Architecture. Morgan Kaufmann (2014) 189–213
9. Sedano, T., Ralph, P., Péraire, C.: Software development waste. In: Proc. of the 39th Int. Conf. on Software Engineering. ICSE '17, IEEE Press (2017) 130–140
10. Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., Carriere, J.: The architecture tradeoff analysis method. In: Proc. of the Fourth IEEE Int. Conf. on Engineering of Complex Computer Systems, 1998. ICECCS '98. (1998) 68–78
11. Kazman, R., Klein, M., Clements, P.: ATAM: Method for architecture evaluation. Technical Report CMU/SEI-2000-TR-004, Carnegie Mellon Sw. Eng. Inst. (2000)
12. Boucké, N., Weyns, D., Schelfthout, K., Holvoet, T.: Applying the atam to an architecture for decentralized control of a transportation system. In Hofmeister, C., Crnkovic, I., Reussner, R., eds.: Quality of Software Architectures. Volume 4214 of Lecture Notes in Computer Science. Springer (2006) 180–198
13. Hunt, A., Thomas, D.: The pragmatic programmer: from journeyman to master. Addison-Wesley (1999)
14. Del Rosso, C.: Continuous evolution through software architecture evaluation: a case study. Journal of Software Maintenance and Evolution: Research and Practice **18**(5) (2006) 351–383

[1] http://www.dimecc.com/